```
MODEL:
! A generic transportation Problem in the LINGO modeling language;
! Given:
    A set of customers, each with a demand amount,
    a set of suppliers, each with a supply capacity, and
    a cost per unit shipped matrix over combinations of suppliers and customers,
   Decide how much to ship from each supplier to each customer, so as to
   Minimize the total cost of shipping,
    subject to,
   Each customer receives the amount demanded, and
   Each supplier ships no more than its capacity;
! Keywords: Demand, Distribution, LINGO, Shipping, Simple, Supply, Transportation model;
SETS:
! The SETS section describes the general data structure;
   SUPPLIER : CAPACITY; ! Each supplier has a capacity;
   CUSTOMER : DEMAND;   ! Each customer has a demand;
! A combination of supplier/customer has a
     cost/unit shipped and amount shipped;
   LINK( SUPPLIER, CUSTOMER): COST, FLO;
ENDSETS
! Here are the data for a specific instance;
DATA:
   SUPPLIER = SUP1 SUP2 SUP3 ;
   CAPACITY =  60   55   51 ;
   CUSTOMER = CUST1 CUST2 CUST3 CUST4 ;
   DEMAND =    35     37    22   32 ;
   COST =       6      2     6    7
                4      9     5    3
                5      2     1    9 ;
 ! You can get the data from, and store results
  to, a spreadsheet with the @OLE() statement,
  essentially, replace "parameter = data,"
  by statements like:  SUPPLIER = @OLE();
!  You must have range names in Excel that match the names
  used in LINGO.
   Similar comments apply
  to a SQL database using the @ODBC() statement;
ENDDATA
! Variables:
    FLO( I, J) = amount shipped from supplier I to customer J;
! The objective;
   MIN = @SUM( LINK( I, J):
                    COST( I, J) * FLO( I, J));

! The capacity constraints. FLO shipped out of I
   must be <= supply at I;
   @FOR( SUPPLIER( I):
 [CAPCON]   @SUM( LINK( I, J): FLO( I, J)) <= CAPACITY( I));

! The demand constraints. Total FLO shipped into J
   must equal demand at J;
   @FOR( CUSTOMER( J):
 [DEMCON]   @SUM( LINK( I, J): FLO( I, J)) = DEMAND;
       );
END
```